

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A system that facilitates the interface of non-integrated applications, comprising:

a processor coupled to memory that retains:

a first application program interface (API) of a first, source-control application, the first API configured as an artifact provider, the first API comprising:

a first web service which returns artifact identifiers comprising universal resource identifiers (URIs) that individually correspond to hosts artifacts of the first source-control application;[[,]]

wherein each artifact of the artifacts of the first source-control application includes including an item[[s]] of data the first application publicly exposed exposes to other applications; and[[,]]

wherein each artifact of the artifacts of the first source-control application is being associated with an artifact type, the artifact type comprising a defect or a build of a software program written in software code stored in the source-control application; and

a first method that receives a URI as a parameter and returns an instance of an artifact that has an artifact identifier that matches the URI that was received as the parameter;

a second application program interface (API) of a second, defect-tracking application, the second API configured as an artifact consumer that hosts artifacts of the ~~second~~ defect-tracking application, the artifacts of the ~~second~~ defect-tracking application including references the ~~second~~ defect-tracking application publicly exposes to other applications, each reference being associated with one referring artifact hosted by the ~~second~~ defect-tracking application and is a link to one referenced artifact of the first source-control application hosted by the artifact provider, the link further comprises a link type that describes a relationship between the referring artifact of the ~~second~~ defect-tracking application and the referenced artifact of the first source-control application, the ~~second~~ defect-tracking application being configured to access the items

of data of the artifacts of the ~~first~~ source-control application via the first API, the ~~first~~ source-control application being configured to access the references of the artifacts of the ~~second~~ defect-tracking application via the second API;

wherein the defect-tracking application is configured to access the items of data of the artifacts of the source-control application via the first API by:

using the web service of the first API to obtain the URIs that individually correspond to artifacts of the source-control application; and

repeatedly calling the first method of the first API with the obtained URIs used as parameters.

2-3. (Cancelled)

4. (Previously Presented) The system of claim 1, further comprising a linking component that links the reference with the corresponding artifact of the first application.

5. (Previously Presented) The system of claim 4, wherein the linking component is an artifact identifier held by the artifact consumer that points to an artifact.

6. (Previously Presented) The system of claim 4, wherein the links is a binary link.

7. (Previously Presented) The system of claim 1, wherein at least one of the provider and the consumer is a tool or service.

8. (Previously Presented) The system of claim 1, wherein the artifact provider registers an artifact type for each artifact it provides, and registers a corresponding link type that each artifact can host.

9. (Original) The system of claim 1, further comprising a generic artifact provider (GAP) that interfaces to a tool to facilitate storing and exposing both artifacts and artifact links.

10. (Original) The system of claim 9, further comprising a GAP adapter that provides an

interface between the GAP and a non-integrated application.

11. (Original) The system of claim 1, further comprising a cache that stores the artifacts and associated artifact links.

12. (Original) The system of claim 1, further comprising a user interface that facilitates presenting inter-artifact references.

13. (Previously Presented) A computer readable storage medium having stored thereon computer executable instructions for carrying out the system of claim 1.

14-16. (Cancelled)

17. (Previously Presented) The system of claim 1, wherein the link is an artifact identifier that is an immutable and uniquely constructed key.

18. (Previously Presented) The system of claim 1, further comprising a link manager that manages a cache by updating and purging cache contents.

19. (Previously Presented) The system of claim 1, wherein the artifact provider and artifact consumer are at least one of loosely coupled and tightly coupled.

20. (Previously Presented) The system of claim 1, further comprising a classifier that makes an inference based on parameters related to at least one of the artifact consumer, artifact provider, and non-integrated applications.

21. (Previously Presented) The system of claim 1, wherein the artifact provider creates and reveals a URI for at least one of loosely-coupled server-based interactions, loosely-coupled clients, caching, and tightly-coupled interactions that support artifact-specific functions by contract with a caller.

22. (Cancelled)

23. (Currently Amended) A computer-readable storage medium having computer-executable instructions for performing a method for facilitating an interface between non-integrated applications, the method comprising:

providing a first application program interface (API), the first API configured as an artifact provider that communicates with a first non-integrated application, the first API comprising:

a first web service which returns artifact identifiers comprising universal resource identifiers (URIs) that individually correspond to artifacts of the first application; and

a first method that receives a URI as a parameter and returns an instance of an artifact that has an artifact identifier that matches the URI that was received as the parameter;

exposing a referenced artifact hosted by the first application via the artifact provider, the referenced artifact comprising an item of public data of the first application;

providing a second application program interface (API), the second API configured as an artifact consumer that communicates with a second non-integrated application, the second application including a referring artifact comprising an item of public data of the second application;

exposing a reference held by second application and the referring artifact associated with the reference via the artifact consumer; and

linking the referring artifact to the referenced artifact via the reference, the reference including an artifact identifier of the referenced artifact, the second application being configured to access the item of public data of the referenced artifact via the first API, the first application being configured to access the item of public data of the referring artifact via the second API;

wherein the second application is configured to access the item of public data of the referenced artifact via the first API by:

using the web service of the first API to obtain a URI that corresponds to the referenced artifact; and

calling the first method of the first API with the obtained URI used as a parameter.

24. (Previously Presented) The method of claim 23, further comprising the acts of:

registering an artifact type for the referring artifact and the referenced artifact; and
registering a link type that the referring artifact and the reference artifact hosts.

25. (Previously Presented) The method of claim 23, further comprising presenting dependency information of the referenced artifact to a user, the information including at least one of link type, artifact type, artifact name, and modification date.

26. (Previously Presented) The method of claim 23, wherein at least one of the artifact consumer or artifact provider is a web service.

27. (Original) The method of claim 23, further comprising generating an artifact proxy that represents data stored in a non-integrated application.

28. (Previously Presented) The method of claim 23, wherein the referenced artifact and referring artifact are representative of at least one of a source file, defect, requirement, test result or build.

29. (Previously Presented) The method of claim 23, wherein linking comprises creating a link between the referring artifact and the referenced artifact that includes a referring DRI, a referenced URI, and a link type.

30. (Original) The method of claim 23, further comprising discovering which referring artifacts hold links to a specific referenced artifact.

31. (Previously Presented) The method of claim 23, further comprising raising an event when the referenced artifact is at least one of created, deleted, and changed.

32. (Previously Presented) The method of claim 23, further comprising providing external addressability for the referenced artifact by the artifact provider.

33. (Previously Presented) The method of claim 23, wherein the first application is a

source control application; and wherein the second application is a defect tracking application.

34-37. (Cancelled)

38. (Currently Amended) A computer-implemented system that facilitates data integration among one or more non-integrated applications in a development environment, comprising:

at least one processor, coupled to a memory, that executes the following computer-executable components:

an integration service in the development environment that includes one or more non-integrated applications that each comprise at least one artifact, the integration service comprises:

a first application and a second application that each include one or more artifacts, the one or more artifacts are items of data of the applications that are publicly exposed, the one or more artifacts include artifact types and unique artifact identifiers;

a first application program interface (API) configured as an artifact provider associated with the first application, the artifact provider that facilitates exposing at least a referenced artifact of the first application, the first API comprising:

a first web service which returns artifact identifiers comprising universal resource identifiers (URIs) that individually correspond to artifacts of the first application; and

a first method that receives a URI as a parameter and returns an instance of an artifact that has an artifact identifier that matches the URI that was received as the parameter;

a second application program interface (API) configured as an artifact consumer associated with the second application, the artifact consumer that facilitates exposing at least a referring artifact of the second application and a reference associated with the referring artifact, the reference includes an artifact identifier corresponding to the referring artifact exposed by the artifact provider, the second application being configured to access the referenced artifact of the first application via the first API, the first application being configured to access the referring artifact of the second application via the second API; and

a linking component that facilitates creation of a link between the referring artifact and the referenced artifact via the reference included in the artifact consumer, the link includes a link type that indicates a type of relationship between the referring artifact and the referenced artifact;

wherein the second application is configured to access the referenced artifact of the first application via the first API by:

using the web service of the first API to obtain a URI that corresponds to the

referenced artifact; and

calling the first method of the first API with the obtained URI used as a
parameter.